

## **APPENDIX B**

### **QUESTIONS SATISFIED BY SPECIFIC MEASURES**

## QUESTIONS SATISFIED BY SPECIFIC MEASURES

SPECIFIC MEASURE	FOCUS QUESTION
<b>1.1.1 Milestone Dates</b>	How frequently has the schedule changed?
	How many activities are scheduled concurrently?
	Is the schedule realistic?
	What is the risk of not delivering on schedule?
<b>1.2.1 Requirements Allocated</b>	Has the requirements test matrix been completed?
	Have all requirements been allocated to at least one design component?
	How many requirements can be directly tested?
<b>1.2.2 Components Designed</b>	Are component designs being completed on time?
	Is the schedule for component designs realistic?
<b>1.2.3 Components Implemented</b>	Are components being completed on time?
	Is the planned implementation rate realistic?
	What components are behind schedule?
<b>1.2.4 Components Integrated and Tested</b>	Is integration and testing being accomplished on schedule?
	Is the planned rate of integration and testing realistic?
<b>1.2.5 Test Cases Completed</b>	Are tests being completed on schedule?
	Is the test schedule realistic?
	What functions have not been tested?
<b>1.2.6 Paths Tested</b>	Have all of the paths been successfully tested?
	How many test cases are required to completely test the software?
	What percentage of the paths have been tested?
<b>1.2.7 Requirements Tested</b>	Are requirements being tested as scheduled?
	Have the tests been successful?
	What functional tests are behind schedule?
	What percent of the functionality has been tested?
<b>1.2.8 Problem Reports Resolved</b>	Are problem reports being closed at an adequate rate?
	Is the rate of problem reporting going down?
	When will testing be complete?
	Which components have the most open problem reports?
<b>1.2.9 Reviews Completed</b>	Are components passing their reviews?
	Are reviews being held on schedule?
	What components have failed their review?
<b>1.2.10 Changes Implemented</b>	How many change requests have impacted the software, the schedule, or the budget?
	Is the rate of change requests decreasing?
<b>1.3.1 Schedule Variance</b>	Are costs conforming to projections?
	How likely is there to be a cost overrun?
	Is the production schedule being met?
<b>1.4.1 Build Component Content</b>	Are components being incorporated as scheduled?
	What components have been added, deferred or eliminated?
	Will each increment contain the specified components?
<b>1.4.2 Build Function Content</b>	Is functionality being incorporated as scheduled?
	What functionality is being deferred?
	Will each increment contain the specified functionality?

## QUESTIONS SATISFIED BY SPECIFIC MEASURES

SPECIFIC MEASURE	FOCUS QUESTION
<b>2.1.1 Effort</b>	Are development resources being applied according to plan?
	Are specific tasks or activities taking more/less effort than expected?
	Is the effort profile realistic?
<b>2.2.1 Staff Allocation</b>	Are certain activities or functions requiring more staff than expected?
	Are sufficient development resources available and allocated for each activity?
<b>2.2.2 Staff Experience</b>	Are sufficient experienced/trained personnel available?
	Will additional training be required?
<b>2.2.3 Staff Turnover</b>	How are experience levels being affected by turnover?
	How many people have been added or left the project?
	What areas are most affected by turnover?
<b>2.3.1 Cost Variance</b>	Are project costs in accordance with the budget?
	What is the projected completion cost?
	What WBS elements or tasks have the greatest variance?
<b>2.3.2 Cost Profile</b>	Are program costs in accordance with budgets?
	Will the project budget be adequate or will there be an overrun?
<b>2.4.1 Resource Availability Dates</b>	Are key resources to be available when needed?
	Is the availability of resources impacting progress?
<b>2.4.2 Resource Utilization</b>	Are the available resources sufficient?
	How effectively are resources being used?

## QUESTIONS SATISFIED BY SPECIFIC MEASURES

SPECIFIC MEASURE	FOCUS QUESTION
<b>3.1.1 Requirements</b>	Are requirements being deferred to later builds?
	Have requirements allocated to each incremental build changed?
	How much has software functionality changed?
	What components are affected by the changes?
<b>3.1.2 Function Points</b>	How much functionality is in the software?
	How much work is to be done?
<b>3.2.1...Lines of Code</b>	Has the size allocated to each build changed?
	How accurate was the size estimate that schedule and effort plans were based on?
	How much has the software changed?
	What components have grown or gotten smaller?
<b>3.2.2 Number of Components</b>	Have components allocated to each build changed?
	How many components need to be implemented and tested?
	How much has the approved software baseline changed?
	Is functionality slipping to later builds?
<b>3.2.3 Words of Memory</b>	Does the memory need to be upgraded?
	How much spare memory capacity is available?
<b>3.2.4 Database Size</b>	How many different data types have to be addressed?
	How much data has to be handled by the system?
<b>3.3.1 CPU Utilization</b>	Have sufficient CPU resources been provided?
	Can the CPU resources support additional functionality?
	Do CPU estimates appear reasonable?
<b>3.3.2 CPU Throughput</b>	Have sufficient CPU resources been Acquired?
	Have large increases in CPU utilization occurred?
<b>3.3.3 I/O Utilization</b>	Do the I/O resources allow adequate data traffic flow?
	Should I/O resources be expanded?
	Can additional data traffic be provided after system delivery?
<b>3.3.4 I/O Throughput</b>	Can the software design handle the required amount of system data in the allocated time?
	Can the software handle additional system data a after delivery?
<b>3.3.5 Memory Utilization</b>	Will the software fit in the processors?
	What is the risk that system errors will be caused by lack of storage space?
	Can the software size increase after system delivery as needed to incorporate new functionality?
<b>3.3.6 Storage Utilization</b>	Have sufficient storage resources been provided?
	What is the expansion capacity?
	Do storage estimates appear adequate?
<b>3.3.7 Response Time</b>	Is the target computer system sufficient to meet response requirements?
	Does the software operate efficiently?
	How long do certain services take?

## QUESTIONS SATISFIED BY SPECIFIC MEASURES

SPECIFIC MEASURE	FOCUS QUESTION
<b>4.1.1 Problem Report Trends</b>	How many problems reports have been written?
	How many problem reports are open? What are their priorities?
	Do report arrival and closure rates support the scheduled completion date of integration and test?
<b>4.1.2 Problem Report Aging</b>	How long does it take to close a problem report?
	Are difficult problems being deferred?
	Are reported problems being closed in a timely manner?
<b>4.1.3 Defect Density</b>	What is the quality of the software?
	What components are candidates for rework?
	What components require additional testing or review?
	What components have a disproportionate amount of defects?
<b>4.1.4 Failure Interval</b>	What is the program's expected operational reliability?
	How reliable is the software?
	How often will software failures occur during operation of the system?
<b>4.2.1 Cyclomatic Complexity</b>	How many complex components exist in this program?
	What components should be subject to additional testing?
	What are the most complex components?
<b>4.2.2 Weighted Methods Per Class</b>	How many complex classes of objects are in this program?
	How many objects should be subjected to additional testing
	What objects are most complex?
	What classes are the most complex?
<b>4.2.3 Response For A Class</b>	Which objects should be subjected to additional testing?
	How many methods are contained within each class?
	Which objects have the most externally invoked methods?
<b>4.2.4 Lack of Cohesion of Methods</b>	Within each class, how many methods reference each variable?
	Which variables are referenced by how many methods?
	How complex are the methods within each class of objects?
<b>4.2.5 Coupling Between Object Classes</b>	Which object classes will be difficult to reuse?
	Which object classes will be most difficult to maintain?
	Which object classes are tightly coupled?
<b>4.2.6 Depth Of Inheritance</b>	Which object classes will be most difficult to test?
	Which object classes will be most difficult to maintain?
	How much code is being reused?
<b>4.2.7 Number Of Children</b>	How much code is being reused?
	Which object classes' methods will require the most testing?

## QUESTIONS SATISFIED BY SPECIFIC MEASURES

SPECIFIC MEASURE	FOCUS QUESTION
<b>5.1.1 Capability Maturity Model Level</b>	Does a developer meet minimum development capability requirements?
	Is the developer's software process adequate to address anticipated program risks, issues, and constraints?
	What project management and software engineering practices can be improved?
	What is the developer's current software development capability rating?
<b>5.2.1 Product Size/Effort Ratio</b>	Is product being developed at a rate to be completed within budget?
	Is the planned software productivity rate realistic?
	How efficiently is software being produced?
<b>5.2.2 Functional Size/Effort Ratio</b>	Is product being developed at a rate to be completed within budget?
	Is the planned software productivity rate realistic?
	How efficiently is software being produced?
<b>5.3.1 Rework Size</b>	How much code had to be changed as a result of correcting defects?
	Is the amount of rework impacting the cost and schedule?
	What was the quality of the initial development effort?
<b>5.3.2 Rework Effort</b>	How much effort was expended on fixing defects in the software product?
	Is the amount of rework impacting cost or schedule?
	What software development activity required the most rework?